Power Analysis Part I

Cryptographic Hardware for Embedded Systems
ECE 3170

Fall 2025

Assoc. Prof. Vincent John Mooney III

Georgia Institute of Technology

Reading

- This lecture covers Chapter 4, “Statistical Characteristics of Power Traces,” of *Power Analysis Attacks: Revealing the Secrets of Smart Cards* by Mangard et al., 2007, ISBN-13: 978-0-387-30857-9, ISBN-10: 0-387-30857-1, e-ISBN-10: 0-387-38162-7
  - Specifically, sections 4.1, 4.2, 4.3 and 4.4
- Georgia Tech has purchased the right for students to download books published by Springer, so you can download a pdf file
- All figures in this lecture are from the aforementioned manuscript

Questions Answered by This Lecture

$2^4 2^8 = 2^{12}$

- If one measures the energy consumption or power of a microchip, what do the power traces reveal?
- What is the statistical methodology used to reveal the information claimed to have been learned?

128 bits

16 8-bit searches

Energy Consumption / Power

- See lecture 22 (the next lecture) and your previous coursework, but the overall result is that energy consumption and power are based in part on actual bit values
- Consider an operation dependent on the most significant bit (MSB) value of a register, e.g., rotate left shift (also called barrel shift)
  - If MSB = 1, shift left and XOR with 0x00000001; o.w., shift left one bit
- Such an operation will have different power traces with many ones versus few

0x10000001

0x00000011

Execution Time

- Consider a processor without a barrel shift assembly instruction
- To implement barrel (rotate) left shift, assembly code must be written to test the MSB and, if the MSB is a one, exclusive-or a one after (non-rotate) left shift by one
- Such code will have data-dependent delay which can be measured

Differential Power Analysis (DPA)

- Consider an encryption chip with an embedded key not revealed, but the hardware is in the possession of an adversary
- Idea: if a key-dependent energy consuming operation can be isolated and executed 1000 times, perhaps the power traces can be added and subtracted appropriately based on (i) knowledge of the exact plaintext values input to encryption and (ii) a guess of the key value
- Notice that a substitution box (S-BOX or just “S”) in AES takes in an 8-bit value and outputs an 8-bit value
- With a power model, predict the power trace and energy consumption values, then make statistical comparisons

Key Insight (pun intended)

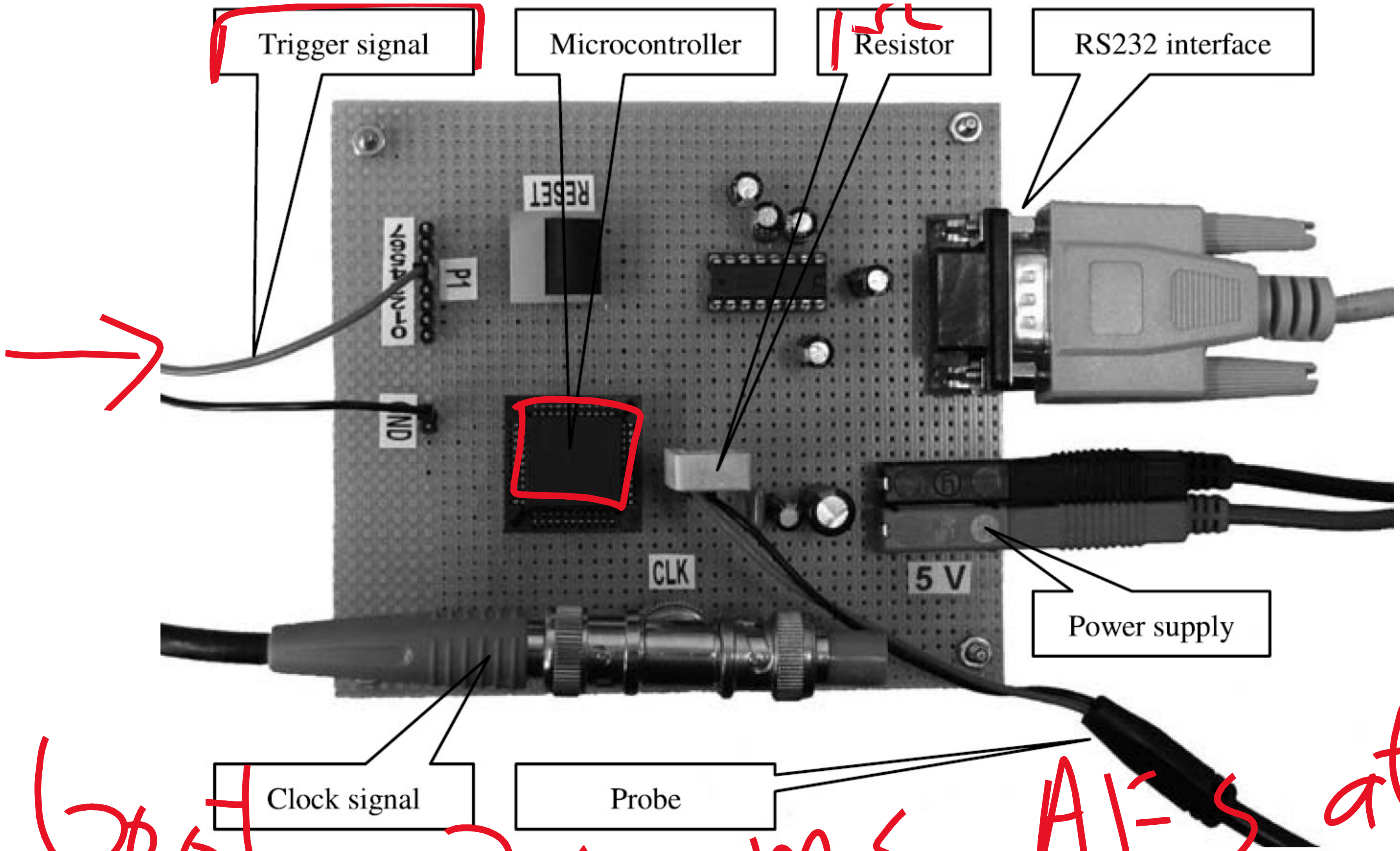
- Since 8 bits of the key are used per S-BOX, 256 experiments can reveal 8 bits of the key
- For a 128-bit key, 16 sets of these 256 experiments can reveal the full 128 bits
- This is 16*256 which is much smaller than 2^{128}
- The hardware does **not** in general enforce 128 bits to be considered as a unit, on the contrary…

Power Analysis

- Cryptographic operations executed on microchips exhibit variations in energy consumption / power
  - Note that power is a rate of energy consumption, i.e., joules per second
- The energy consumption of microchip implementations of cryptography depends on the data values (P_{data}) and the specific operations (typically mathematical or memory storage) performed (P_{op})

99.99%

- There is also an electrical noise component ($P_{el.\ noise}$) and constant (P_{const})
- $P_{total} = \boxed{P_{op}} + \boxed{P_{data}} + \boxed{P_{el.\ noise}} + P_{const}$
- This model may be refined for certain situations, but suffices for most
- Note that cryptographic information may be revealed by P_{op} and P_{data}



boot 200 ms AES after

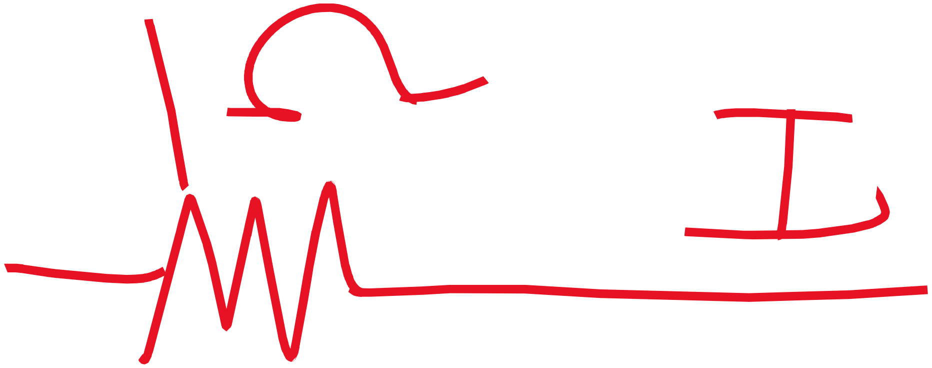
Figure 3.9. Picture of the measurement setup for the attacks on the 8-bit microcontroller.

Example Power Measurement Setup

- The previous slide shows the microcontroller power measurement setup of Mangard et al.
- Microcontroller power supply is 5V and clock frequency 11 MHz
- Voltage drop across a 1 Ω resistor connected to the power supply (Gnd) is measured by an oscilloscope
- Oscilloscope can sample 8 bits of resolution every nanosecond (GHz)
- Typically measure every four ns in experiments (250 Million Samples per second or MS/s)

Consider a Single Point in a Power Trace

- A moment in time
- Aim to determine the probability distributions of P_{op} , P_{data} and $P_{el.\ noise}$



Power Supply

$P = IV$

$P \sim V$

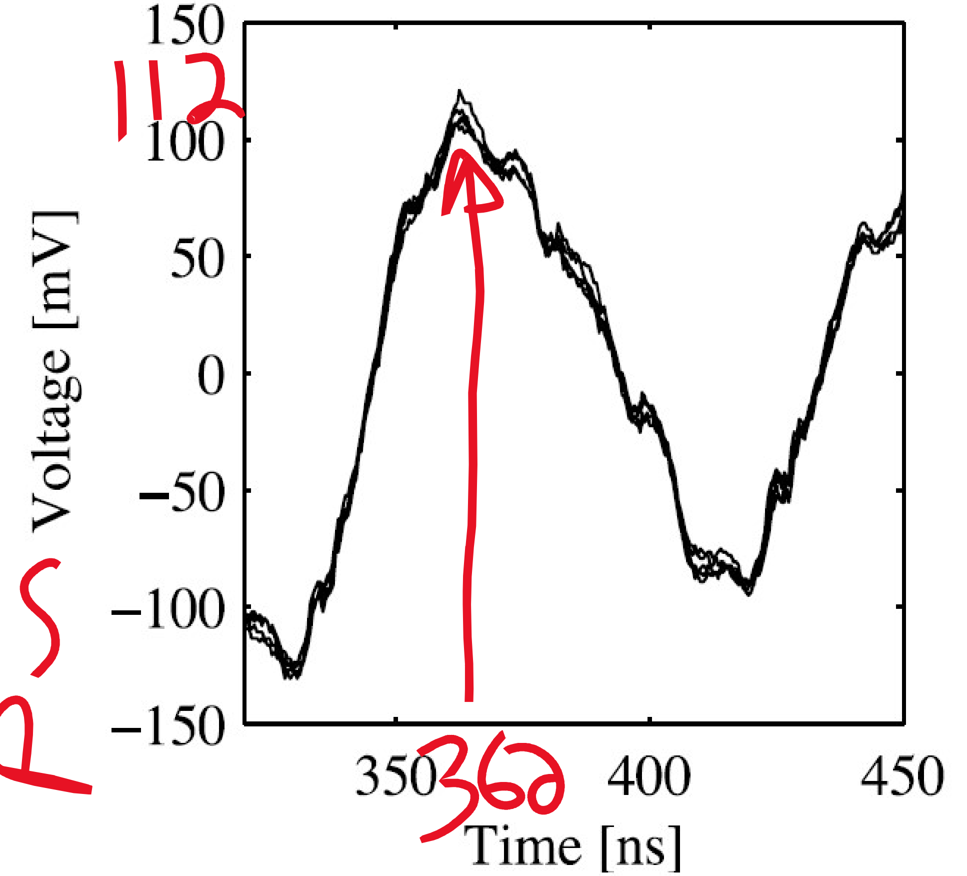


Figure 4.1. Power traces look very similar if the same data is processed.

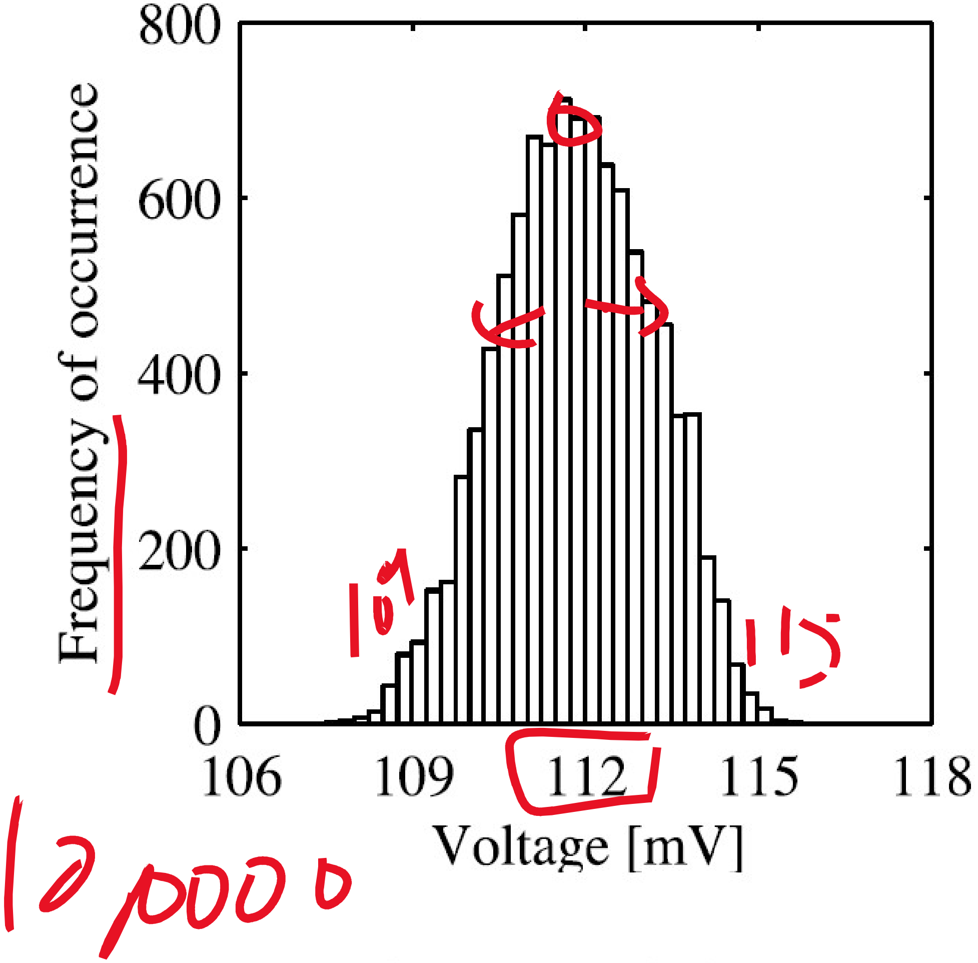


Figure 4.2. Histogram of the power consumption at 362 ns of Figure 4.1.

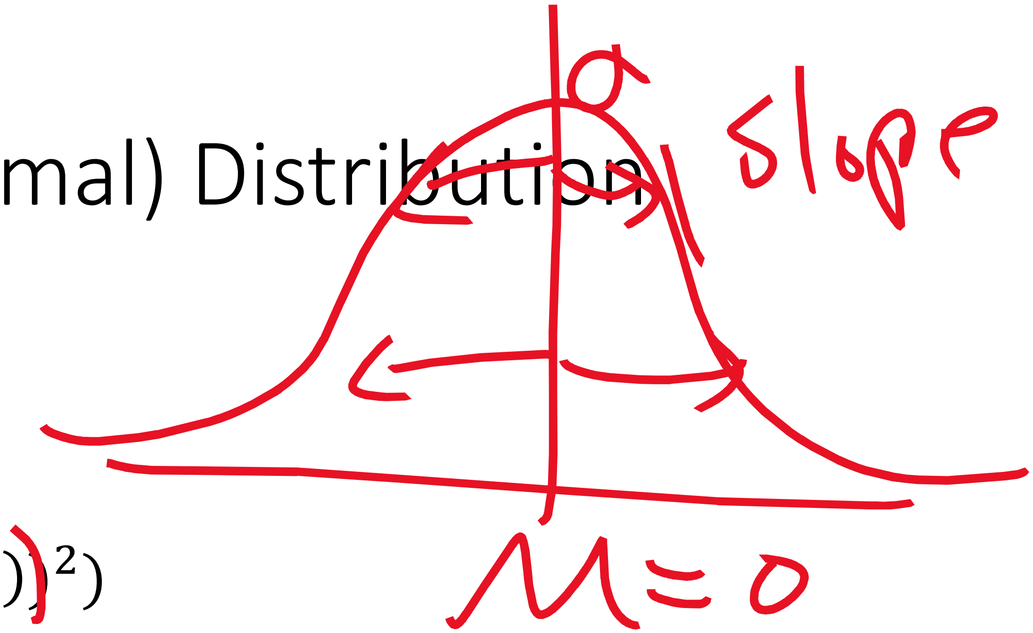
Figures 4.1 and 4.2 from Mangard et al.

- Fig. 4.1 shows five power traces with the same data and instruction (a load of a byte of data with value zero from on-chip memory to a register)

  ld ∅

- The differences in the power traces are due to noise
- Fig. 4.2 shows a histogram of 10,000 power traces of the same operation considering the Voltage across the resister (recall P=IV and I is typically a constant)
  - Most of the measurements are near 112 mV
  - Very few measurements are below 109 mV or above 115 mV

Gaussian (a.k.a. Normal) Distribution



- $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$
- $\mu = E(X)$
- $\sigma^2 = Var(X) = E((X - E(X))^2)$
- $X \sim N(\mu, \sigma)$ means that X is normally distributed (has a Gaussian distribution) with mean value μ and standard deviation σ
- The "standard normal distribution" has μ = 0 and σ = 1

$\sigma = \sqrt{Var}$

Back to Our Experiment at 362 ns

~~le2~~ zero

$n = 10{,}000$

- We can estimate $\mu = E(X)$ with the average $\bar{x}$ calculated empirically:
  - $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

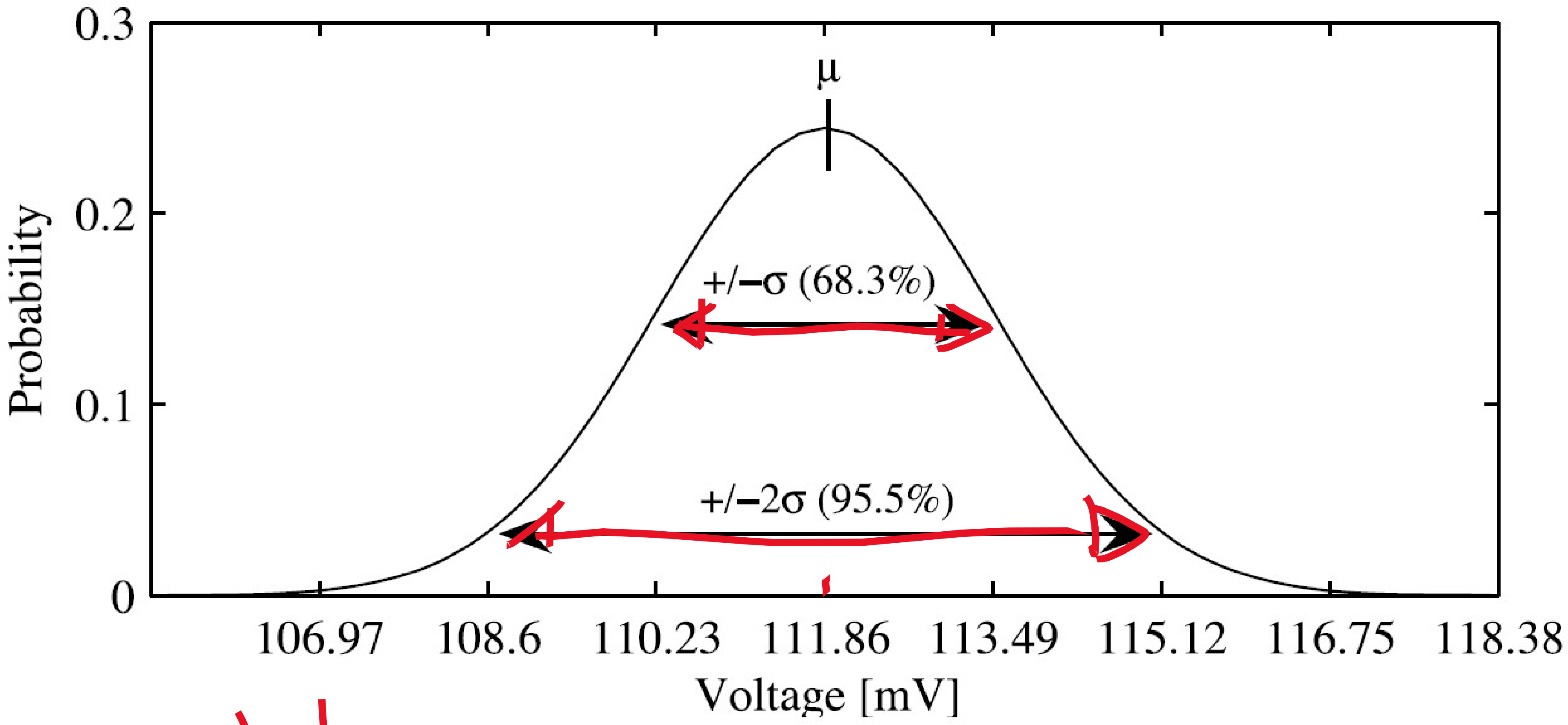
$\bar{x}$

- We can estimate $\sigma = \sqrt{Var(X)}$ with the square root of the variance also calculated empirically:
  - $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$

4

- For this experiment, the following is calculated from the 10,000 traces
  - $\mu = \bar{x} = 111.86\,mV$
  - $\sigma = s = 1.63\,mV$
  - Thus, $X \sim N(111.86, 1.63)$ where the units are millivolts

$X \cong N(\mu, \sigma)$



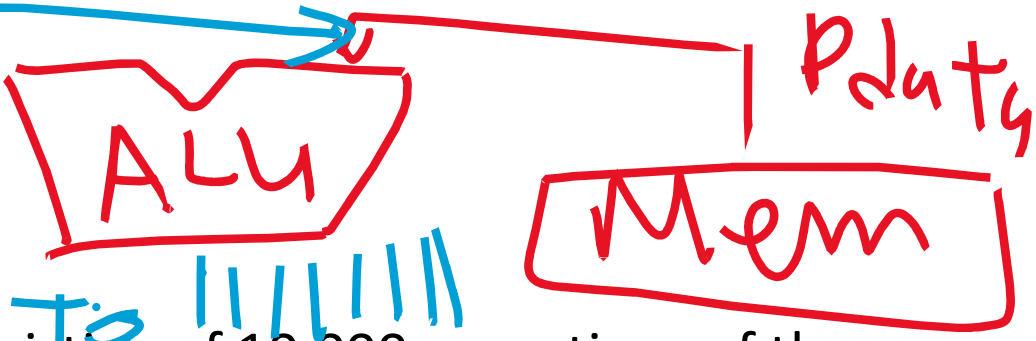
std

Figure 4.3. The normal distribution $\mathcal{N}(111.86, 1.63)$ models the power consumption at 362 ns

©Georgia Institute of Technology, 2018-2025

Power Analysis So Far

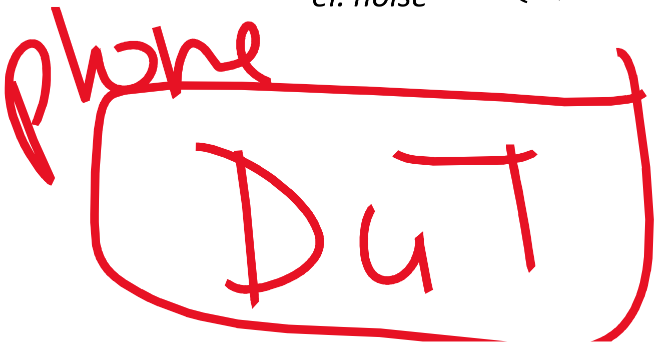
bus value set to 11111111



- From our current experiment consisting of 10,000 executions of the same memory operation with a data value of zero, we find
  - $E(P_{const}) = 111.86\ mV$
  - $E(P_{op}) = E(P_{data}) = E(P_{el.\ noise}) = 0$
  - $Var(P_{const}) = 0$
  - Since the operation executed and the data used do not change, $Var(P_{op}) = Var(P_{data}) = 0$

P

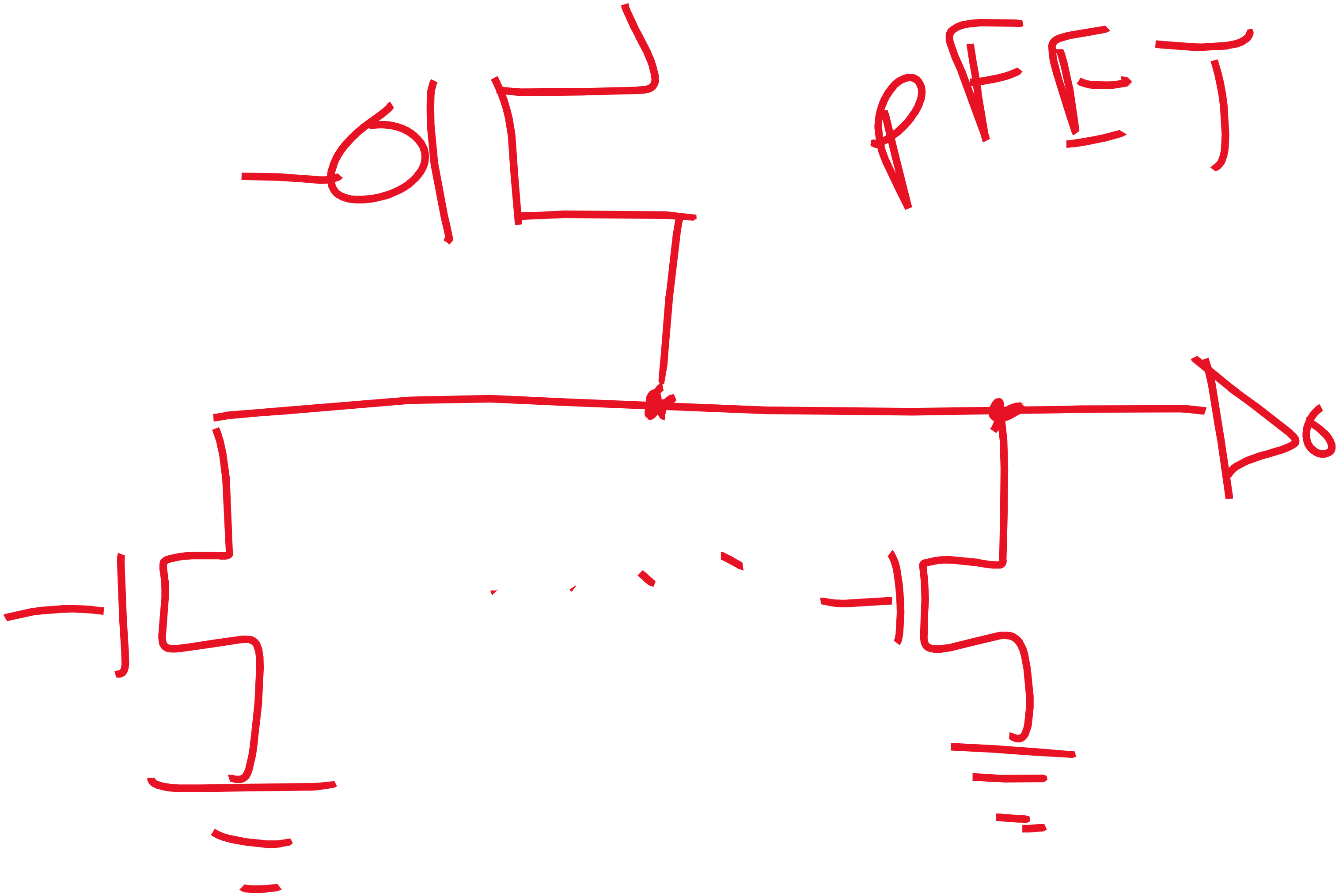
  - $\Rightarrow P_{el.\ noise} \sim N(0{,}1.63)$



same brand,

Next: Data Dependence

- Instead of moving a constant zero from memory to a register location as was done for estimating the distribution of noise on the power pin of this microcontroller, we now vary the eight bit memory data value among all 256 possible values
- 200 measurements for each value = 256* 200 = 51,200 total measurements
- Again, let us look at what happens at 362 ns
- (Quick note before we continue: this discussion will assume large amounts of encrypted data values which are evenly distributed among all possibilities, e.g., zero does not occur more often than any other number; another way to say this is that data values are uniformly distributed)


pFET

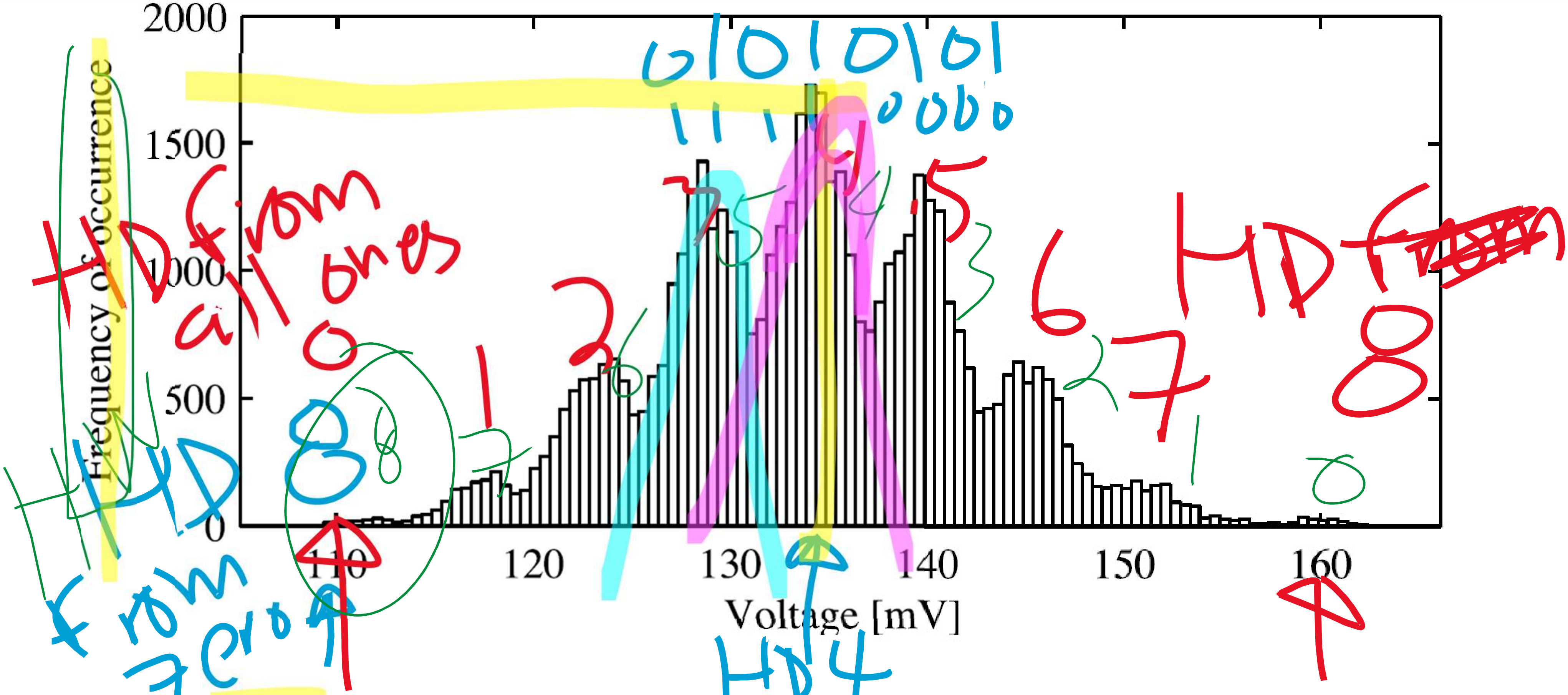


Figure 4.4. Histogram of the power consumption at 362 ns if different data values are transferred from the internal memory to a register.

Examples

bit value	HW	HD from 11111111
00000000	0	8
01000000	1	7
00110000	2	6
⋮	⋮	⋮

Histogram Comments on Figure 4.4

- A precise description (close to exact) is not possible with a single Gaussian
- However, upon further inspection, there do appear to be nine Gaussians
- Why could this be?
  - Answer: consider the Hamming Weight (HW), i.e., the weight (counted in bits) between each possible data value and zero
- *NOTE: Mangard et al. define HW as the Hamming distance between a binary data value and zero (i.e., the same number of bits but all zeros)*
  - *For a Byte, the possible HW values are 0, 1, 2, 3, 4, 5, 6, 7 and 8*

HW Calculations on 8-bit Values

- HW 0: 00000000 (TOTAL: 1) All
- HW 1: 00000001, 00000010, …, 10000000 (TOTAL: 8) All-1
- HW 2: 00000011, 00000101, …, 10000001, 100000010, …
- … HW 4
- HW 7: 11111110, 11111101, …, 01111111 (Total: 8) All-7
- HW 8: 11111111   (TOTAL 1) None
- Result is a binomial distribution
  - Probability of HW 4 is 27.3%, HW 3 or 5 have an equal probability of 21.9%, …, HW 1 or 7 have a probability of 3.125%, HW 0 or 8 have probability 0.39%

Explanation for Figure 4.4

- Function of HW
  - The greater the Hamming distance from 0b1111111 the greater the energy consumption P_{data}
  - Add in $P_{el.\ noise}$
- Load and store instructions in any ISA are known to have a large HW energy consumption component due to the way SRAMs are designed
  - Rows and columns have to be charged and discharged in an array of 6T single bit memory elements
  - Sense amplifiers are used to detect when a 1 is overpowered by a 0 and vice-versa
  - Furthermore, for this microcontroller, bus lines are precharged to 1 each time, so case 0b11111111 (HW 8 measured from zero) has the least power

Question: How to Remove the Noise?

- Calculate the mean voltage for each HW
- In this example, find 111.9, 117.6, 123.2, 128.7, 134.0, 139.5, 145.1, 151.2 and 159.6 (all in mV)
- Earlier we found that $P_{el.\ noise} \sim N(0, 1.63)$, i.e., we know that $\sigma = s = 1.63\ mV$

Question: How to Remove the Noise?

- Calculate the mean voltage for each HW
- In this example, find 111.9, 117.6, 123.2, 128.7, 134.0, 139.5, 145.1, 151.2 and 159.6 (all in mV)
- Earlier we found that $P_{el.\ noise} \sim N(0, 1.63)$, i.e., we know that $\sigma = s = 1.63\ mV$
- Answer: we **cannot** remove the noise
  - This is due to the fact that the noise has $\mu = 0$, i.e., $E(P_{el.\ noise}) = 0$
- However, we can account for the noise statistically!
  - $\sigma = 1.63\ mV$

Result

- Can superpose nine Gaussian distributions to accurately model the measurements

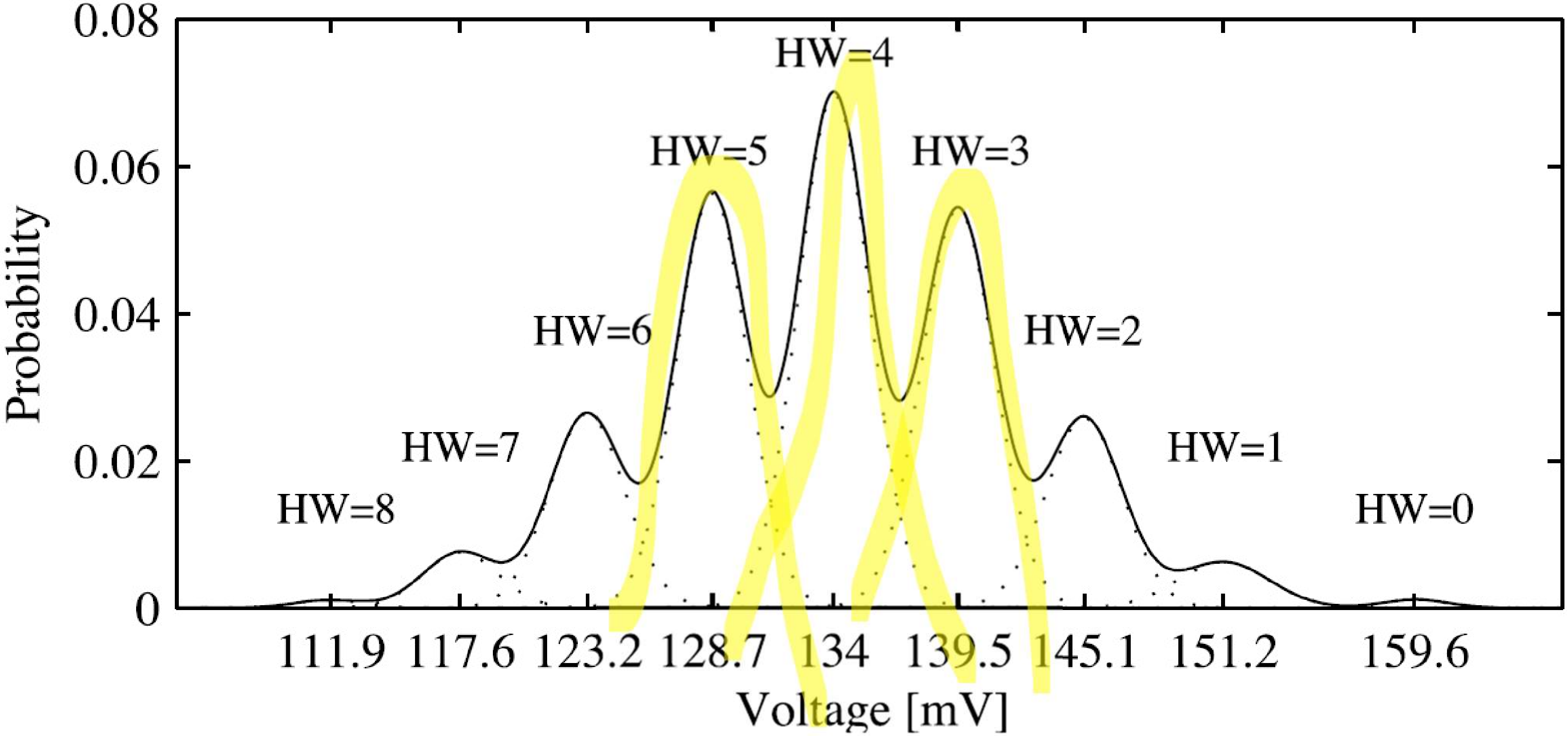


Figure 4.5. The distribution of the power consumption when the microcontroller transfers different data from the internal memory to a register.

Nine Gaussian Distributions

- Mangard et al. propose the following on page 68
  - *$E(P_{const})$* = 134 mV for each of the nine distributions
  - *$E(P_{data})$* = 0 overall (for the combination of the nine distributions)
    - Taken individually, -22.67, -16.92, -11.35, -5.86, -0.49, 4.96, 10.53, 16.68 and 25.12 *mV*
    - Each HW has a Gaussian weighted by the binomial distribution of P_{data}
- Figure 4.4 shows the result where the sum under the curve results in a total of 1 (i.e., the sum of the probabilities sums to 1)

ld instr used
to obtain
Sbox result
⇒ side channel
when key byte input

Next: Energy Consumption per Operation

- Similar to previous, but now alter the ISA operation type
  - Some operations, e.g., load and store or add and subtract, can be grouped together
  - Some operations are very specific, e.g., floating point multiply
  - Also may have to account for multicycle operations
  - As stated in the book, the result is that P_{op} can also be approximated reasonably accurately for this microcontroller (and many other instruction-set architectures or ISAs) by a Gaussian distribution

Signal to Noise Ratio (SNR)

$P_{tota} = P_{data} + P_{op} + P_{el.n.} + P_{con}$

- Two questions: what information is the attacker seeking and what do the points of a power trace provide towards revealing this information?
- We begin with a distinction between exploitable power measurement data $\boxed{P_{exp}}$ which must be due to either P_{data} or P_{op}
  - The aspects of P_{data} and/or P_{op} which may not be exploitable, e.g., due to various bits switching back and forth not under observation are called $P_{switching}$ or $\boxed{P_{sw.\ noise}}$
- Therefore, $\boxed{P_{exp} + P_{switching}} = \boxed{P_{data} + P_{op}}$
- Furthermore, $P_{total} = \underbrace{P_{exp} + P_{switching}} + P_{el.\ noise} + P_{const}$

SNR Continued

to be measured

- $SNR = \frac{Var(Signal)}{Var(Noise)}$
- $SNR = \frac{Var(P_{exp})}{Var(P_{switching} + P_{el.\ noise})}$

Example

- A processor operates on an 8-bit value where each bit is independent and uniformly distributed
- Assume that the value of the second bit is always the complement of the first bit in the experiments carried out
  - E.g., 0b$X_7X_6X_5X_4X_3X_2X_1$0 and 0b$Y_7Y_6Y_5Y_4Y_3Y_2Y_1$1 where the first bit considered in our analysis is the case of the LSB = 0 and the second bit considered in our analysis is the case of the LSB = 1
  - The other 14 bits are independent and uniformly distributed
- P_{exp} consists of the energy consumed by the LSB
- $P_{switching}$ consists of the energy consumed by the rest of the bits

Table

Example (continued)

200 per data value

- We have 51,200 power traces as computed already earlier
- Select the 25,600 traces with LSB = 1
- Figure 4.6 shows the resulting histogram at 362 ns

256

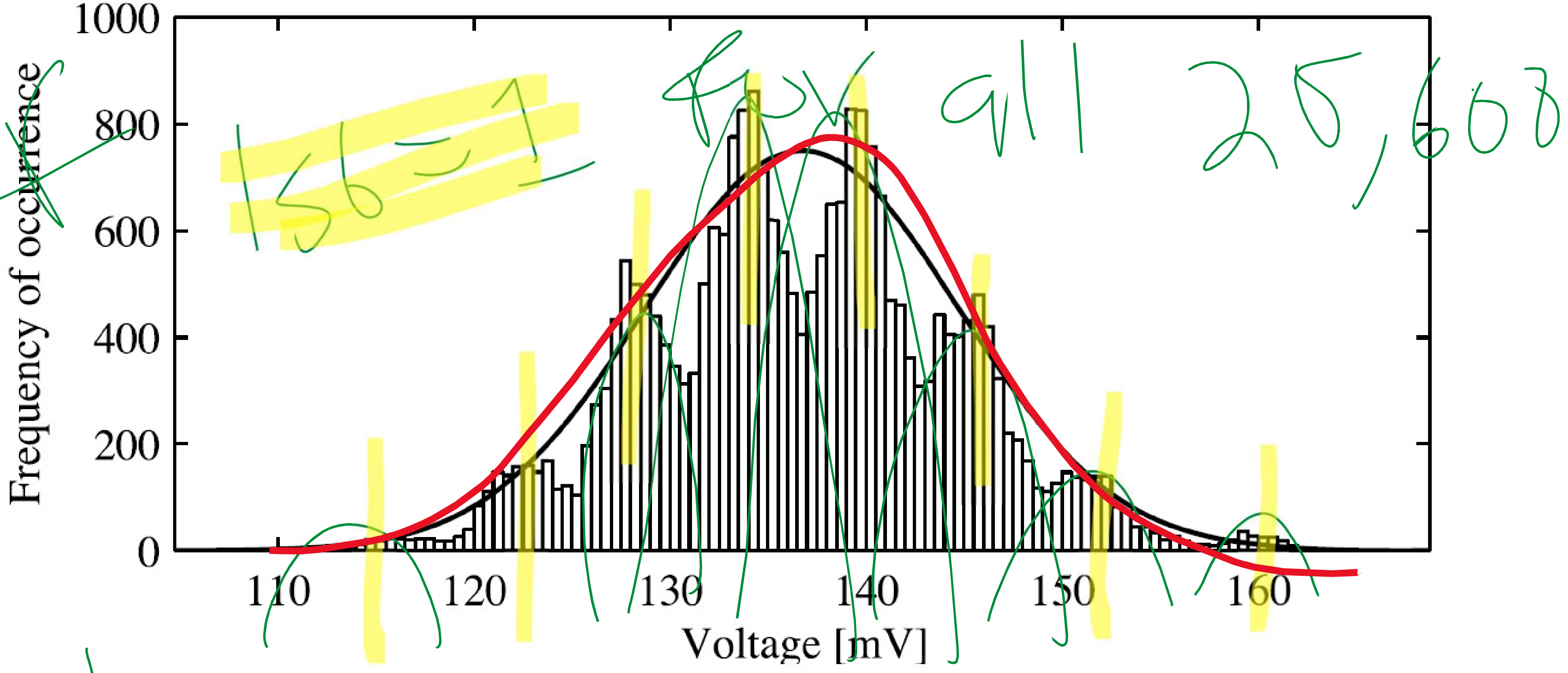
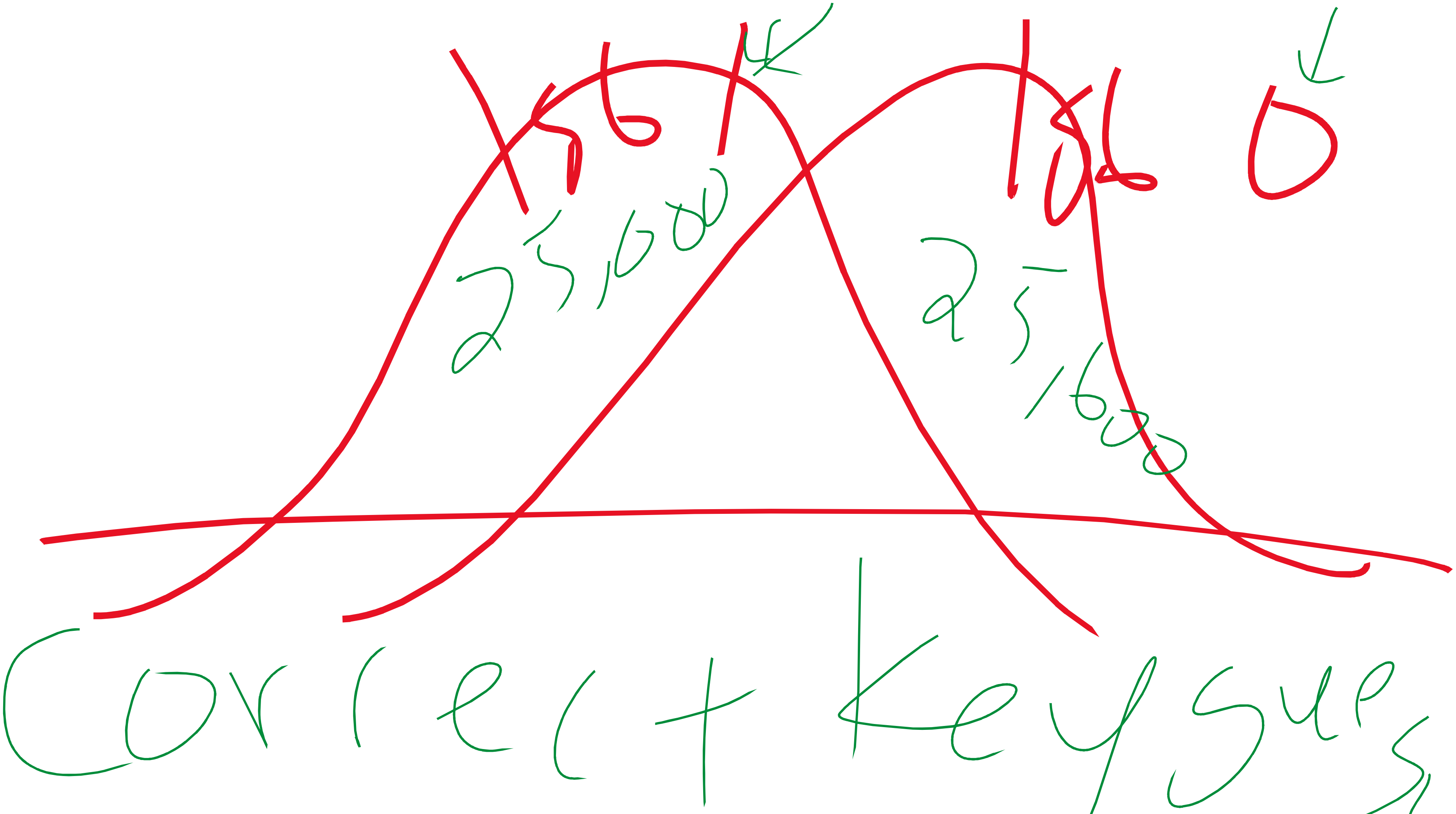
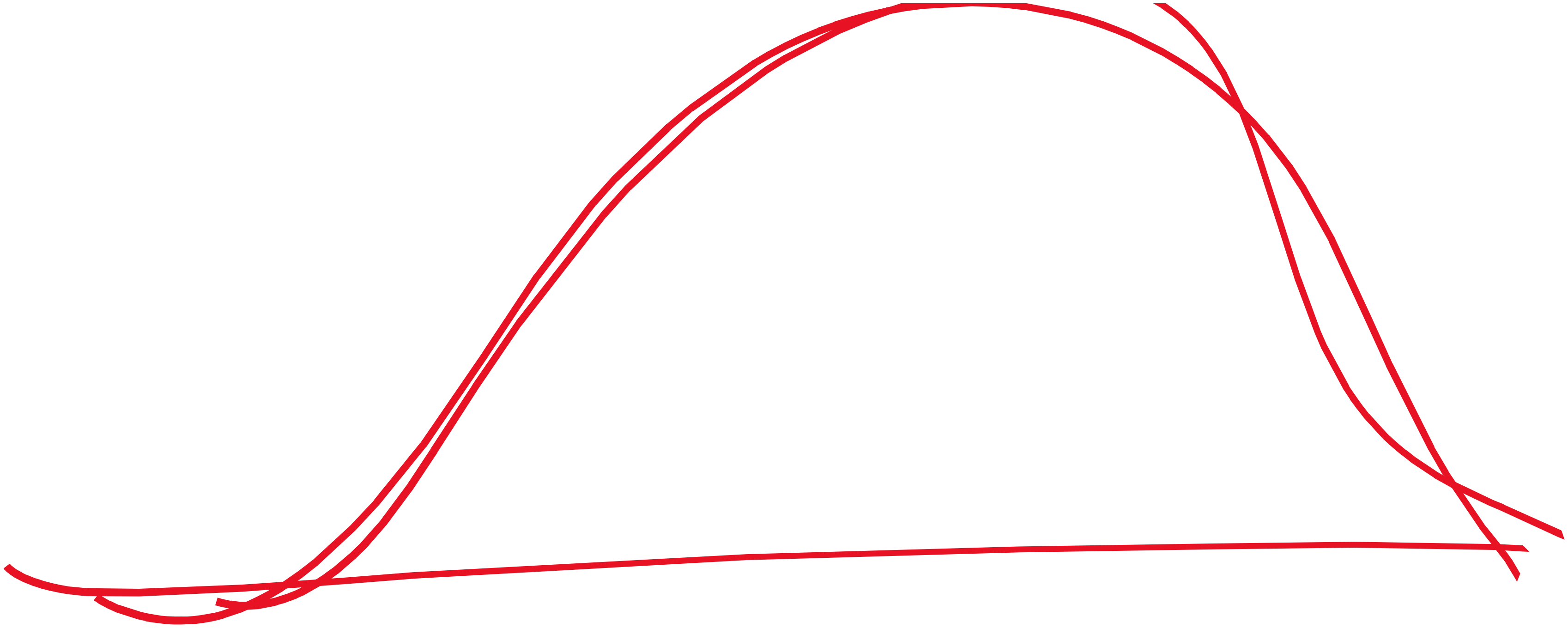


Figure 4.6. Histogram of the total noise ($P_{sw.\ noise} + P_{el.\ noise}$) if the exploitable signal is the LSB of the byte that the microcontroller processes. This noise is approximately normally distributed.

NOTE: Recall that we have defined *$P_{switching}$* to be identical to (i.e., another name for) *$P_{sw.\ noise}$*



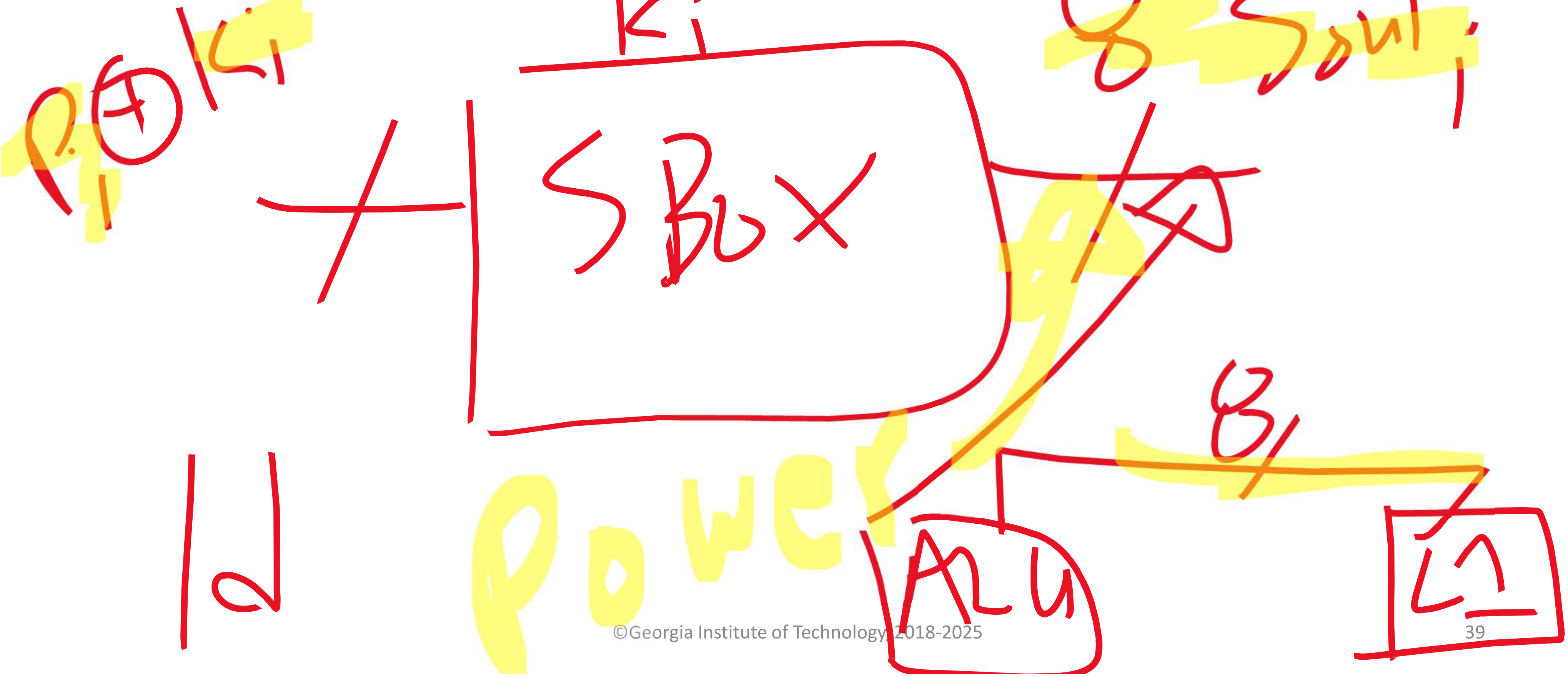
Correct Keyspaces



incorrect

Key suck

$$51{,}200 = \underbrace{256}_{Ki} * \boxed{200}$$



(i) ld instr. corr. Sbox

(ii) plaintext p_i

(iii) gues k_i ✓ correct if

(iv) from 51,200 file, predict LSbit into (0) LSB=0

(1) LSB=1

(v) if (0) − (1) statist. sig.

empi std dev

Comments

- $P_{switching}$ in Fig. 4.6 has a binomial distribution
- Fig. 4.6 also includes $P_{el.\ noise}$
- However, we can approximate Fig. 4.6 with a single Gaussian as shown, in particular since we assume data is uniformly distributed
- The result is $\sigma = s = 7.54\ mV$; thus, for the LSB (1-bit) scenario, we find that σ of $P_{switching} + P_{el.\ noise}$ = 7.54 mV
- Hence, Var($P_{switching} + P_{el.\ noise}$) = (7.54 mV)^2 = 56.85 mV^2
- Also, earlier we found that σ of $P_{el.\ noise}$ for one bit = 1.63 mV
- Hence, Var($P_{el.\ noise}$) = (1.63 mV)^2 = 2.67 mV^2

even only 8 HWs (inst. of 9)

PAA, p3.75

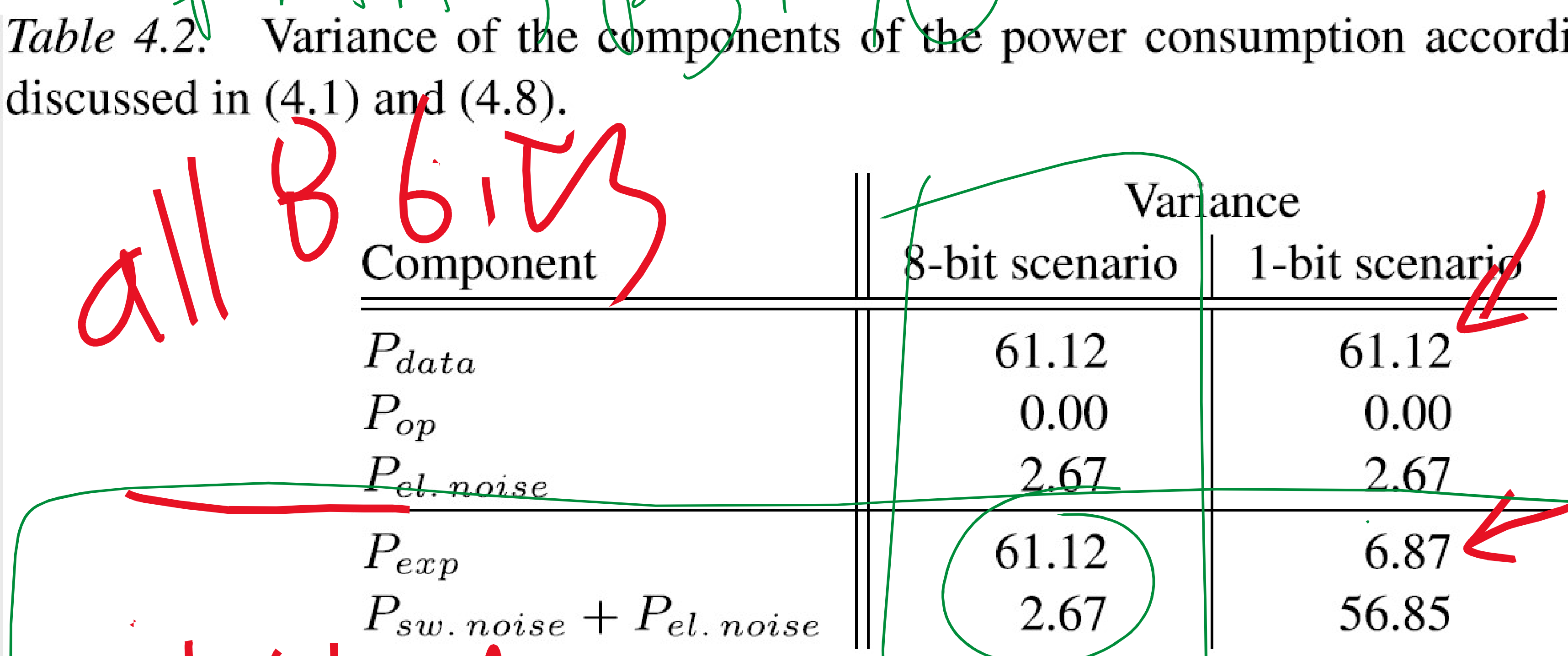


Table 4.2. Variance of the components of the power consumption according to the models discussed in (4.1) and (4.8).

Component	Variance	
	8-bit scenario	1-bit scenario
P_{data}	61.12	61.12
P_{op}	0.00	0.00
$P_{el.\,noise}$	2.67	2.67
P_{exp}	61.12	6.87
$P_{sw.\,noise} + P_{el.\,noise}$	2.67	56.85

all 8 bits

major

expl var

just 1, e.g., LS bit = $\frac{6.87}{56.85}$

SNR

Comparison

- SNR is much higher for 8-bits than for 1-bit

$$SNR = \frac{Var(signal)}{Var(noise)} = \frac{61.12}{2.67}$$

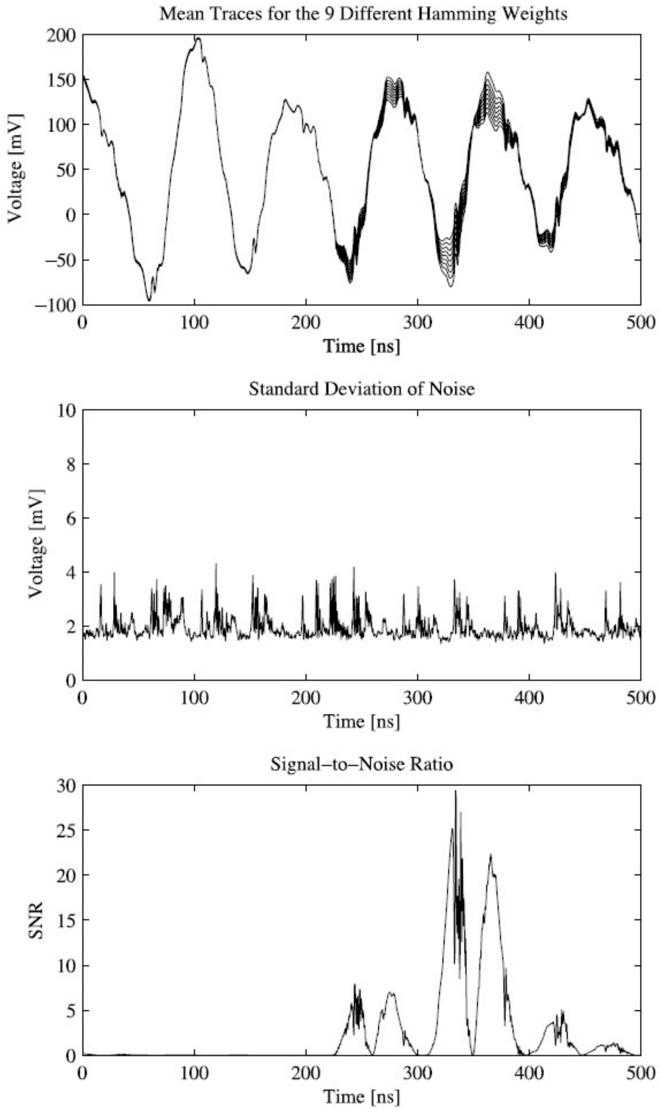
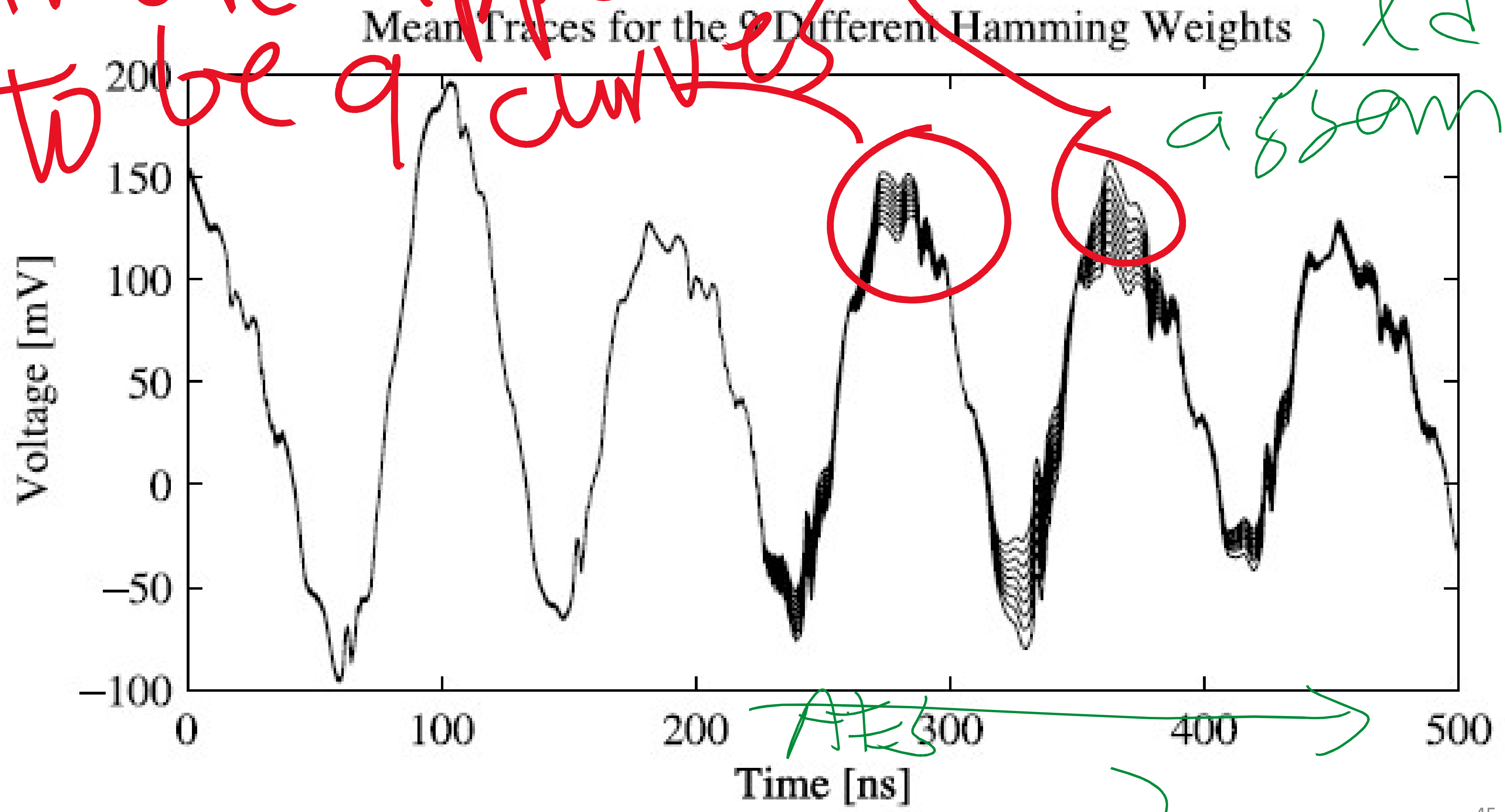


Figure 4.7. The signal levels, the standard deviation of the noise, and the SNR when attacking a uniformly distributed 8-bit data value on our microcontroller.

There appear to be 9 curves

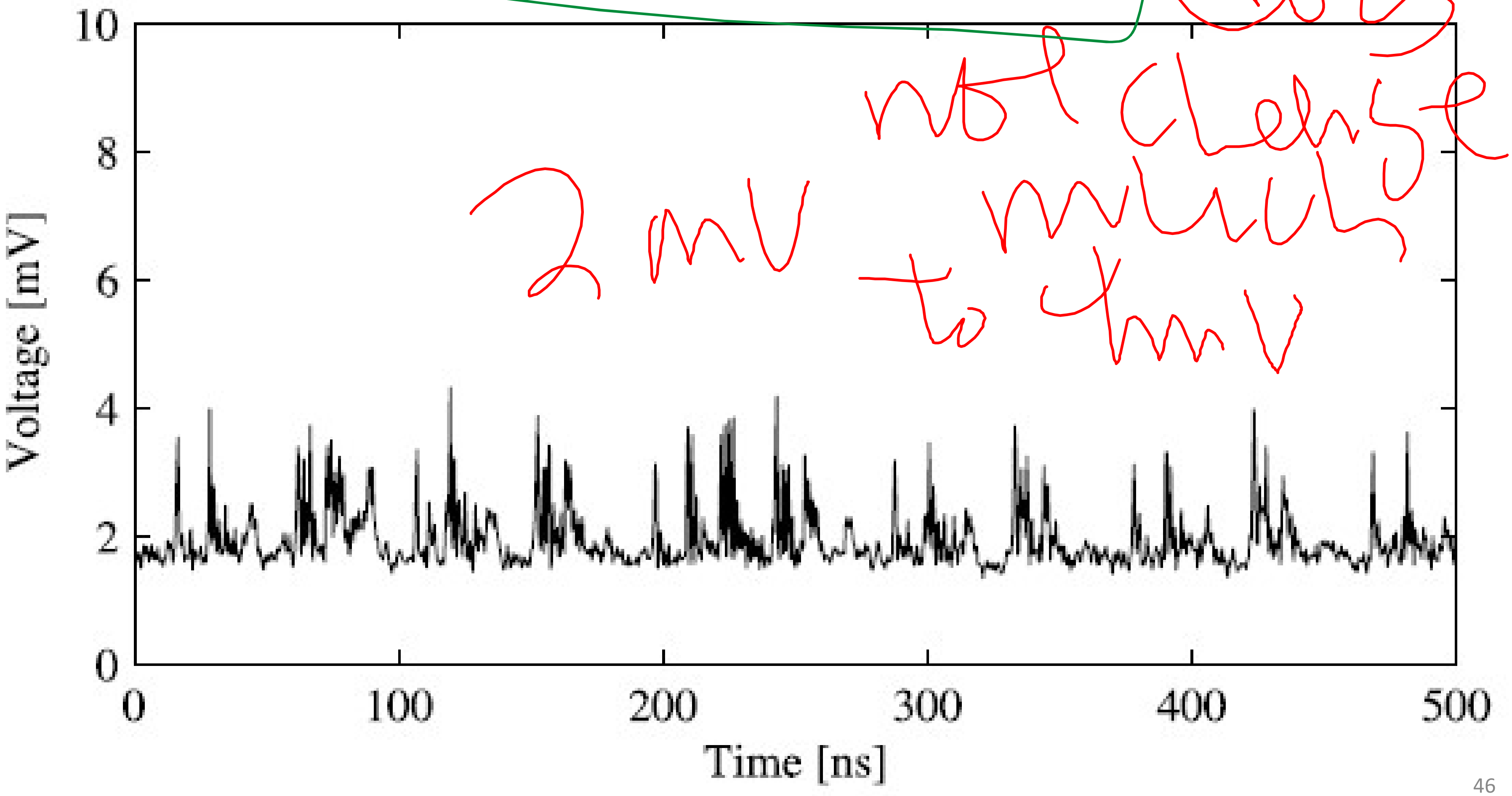

Mean Traces for the 9 Different Hamming Weights

ld assembly

AES

Time [ns]

Standard Deviation of Noise



does
not change
2mV much
to 4mV

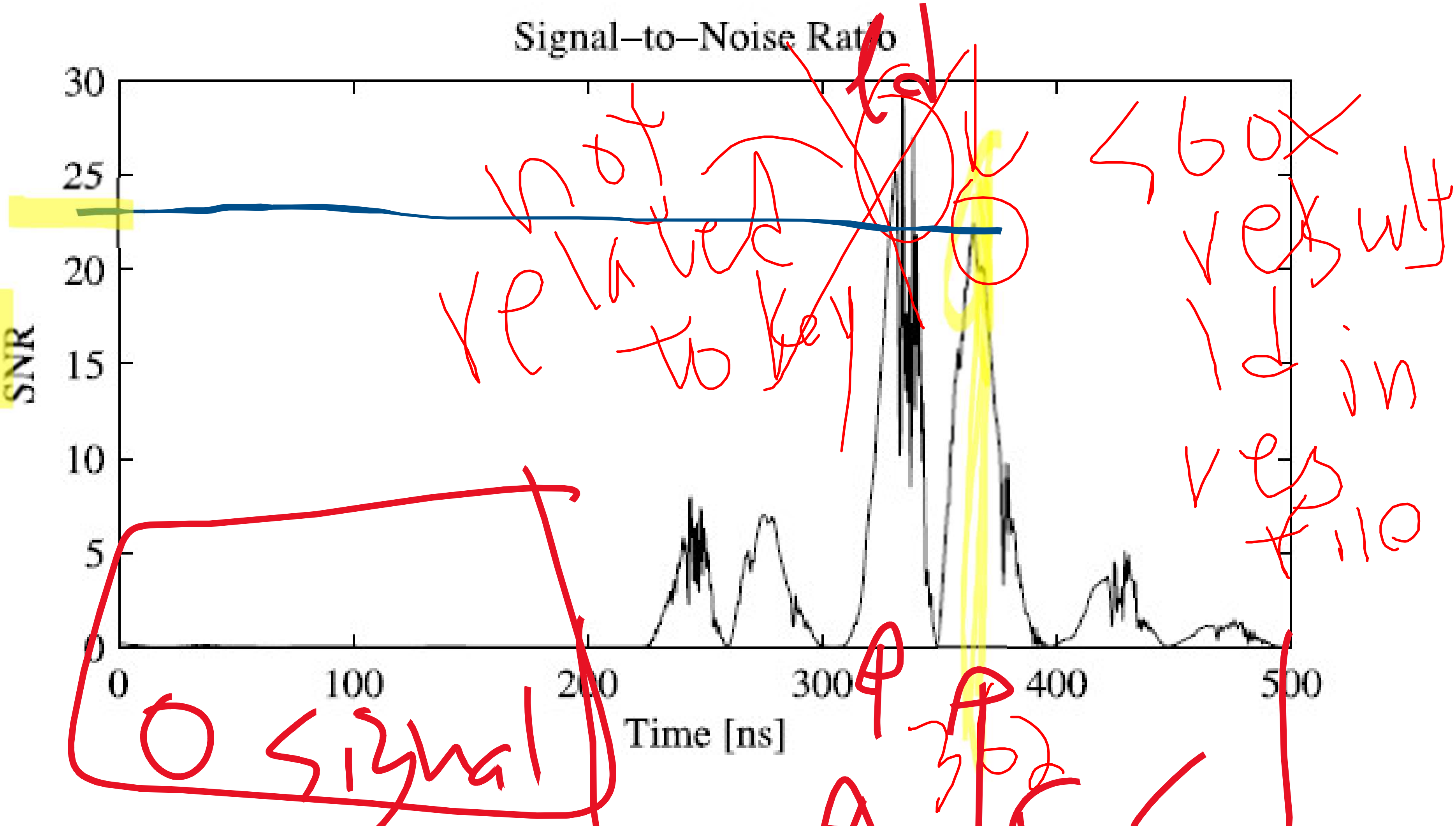


Figure 4.7. The signal levels, the standard deviation of the noise, and the SNR when attacking a uniformly distributed 8-bit data value on our microcontroller.

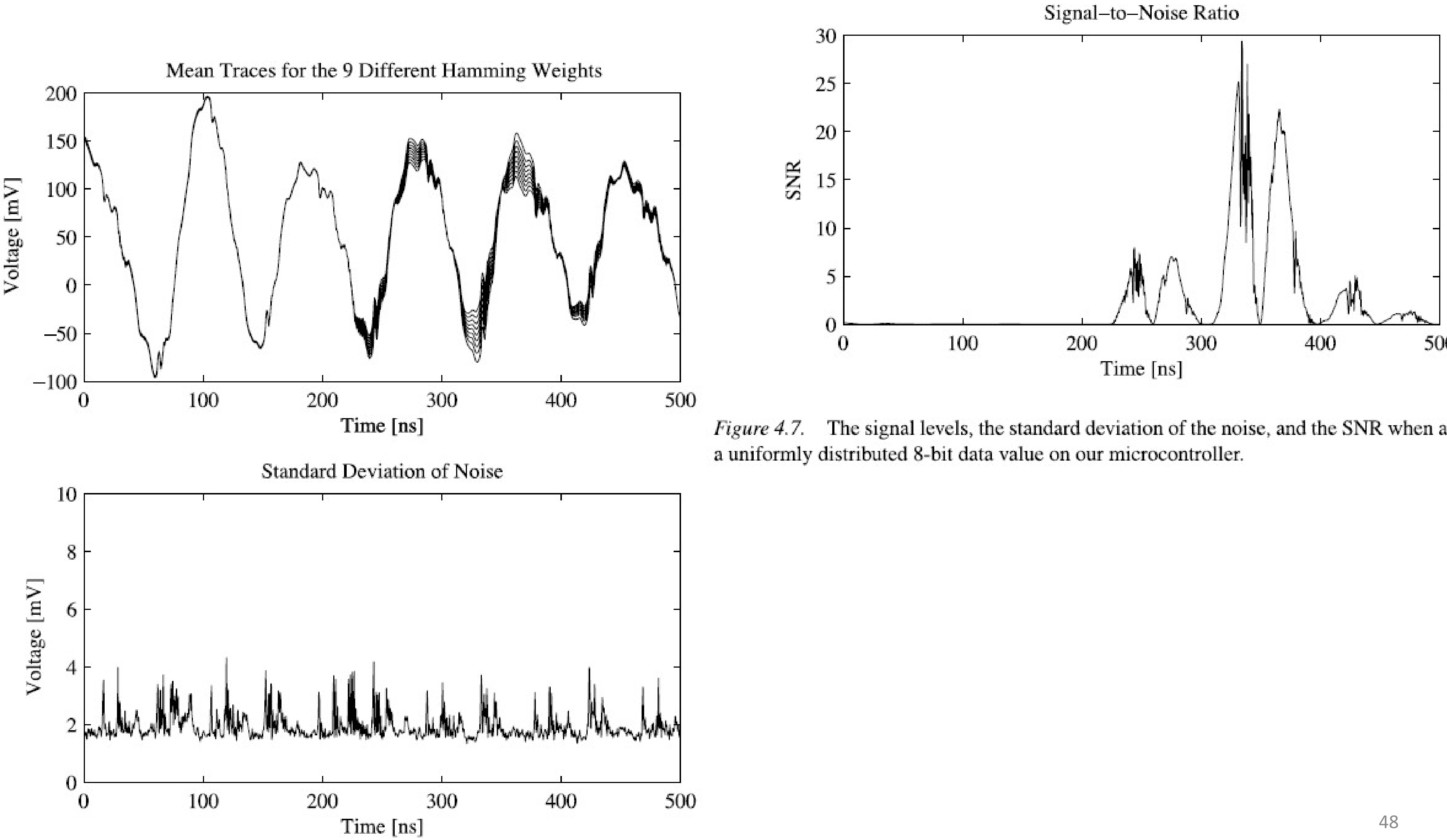


Figure 4.7. The signal levels, the standard deviation of the noise, and the SNR when attacking a uniformly distributed 8-bit data value on our microcontroller.

Comments

4.7(a) shows Var = (std. dev)^2 not only at 362 ns but many pts. Var(pexp) betw. 200ns + 500nto

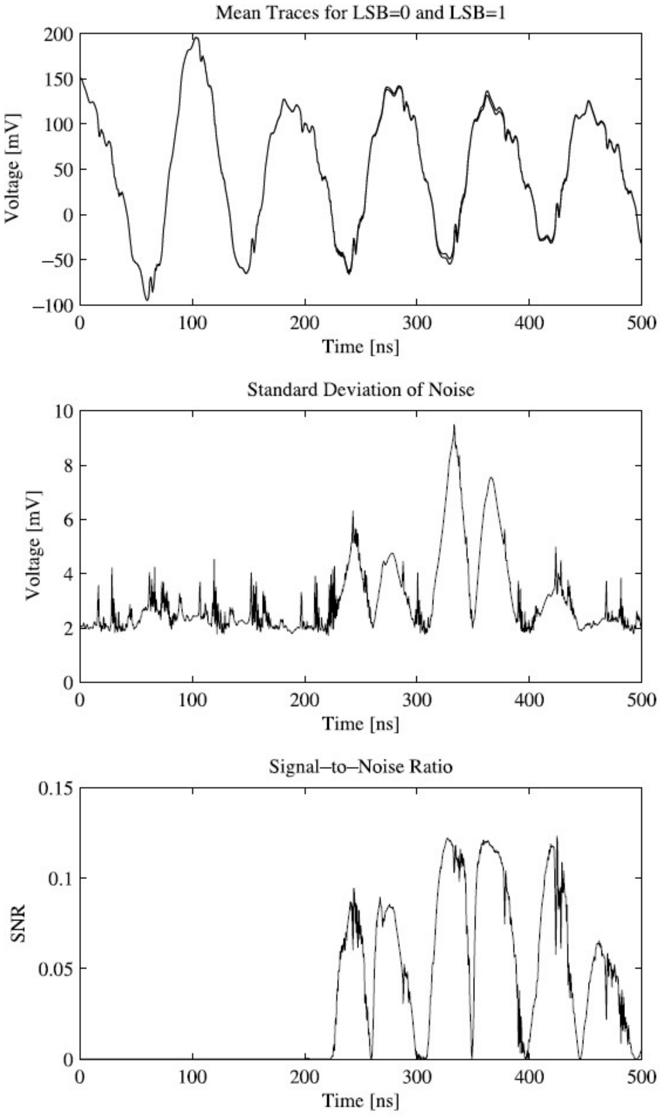
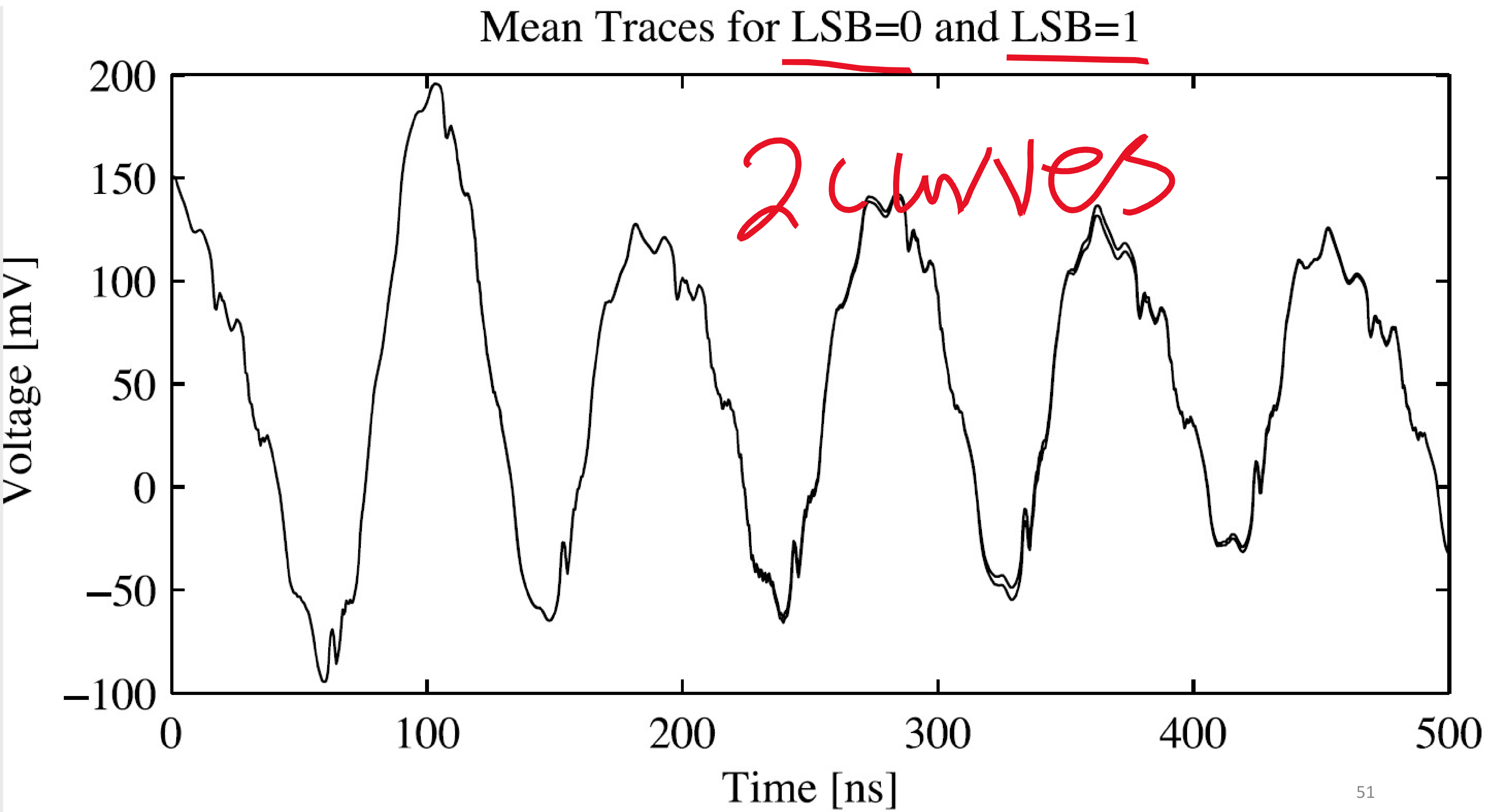
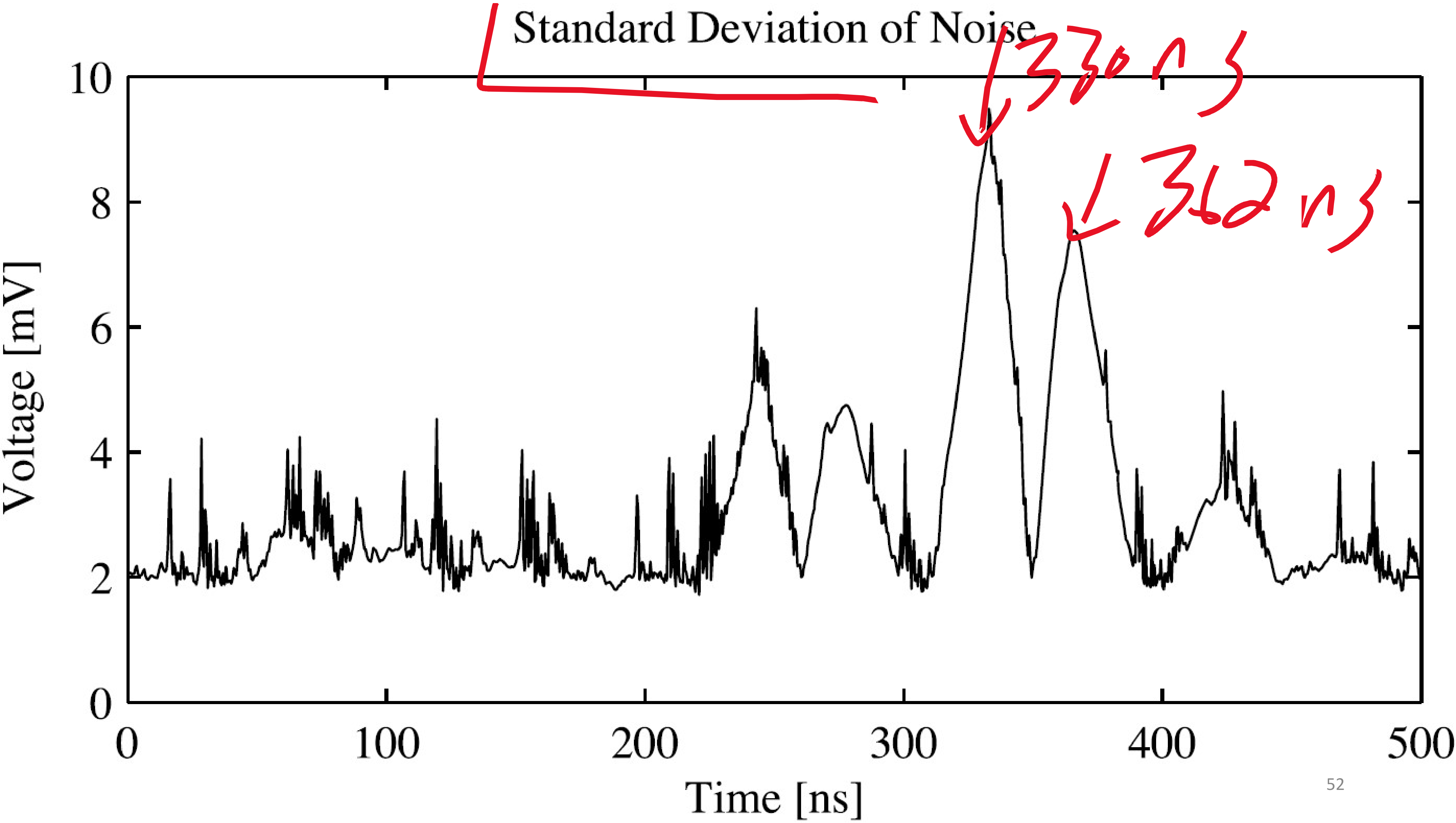


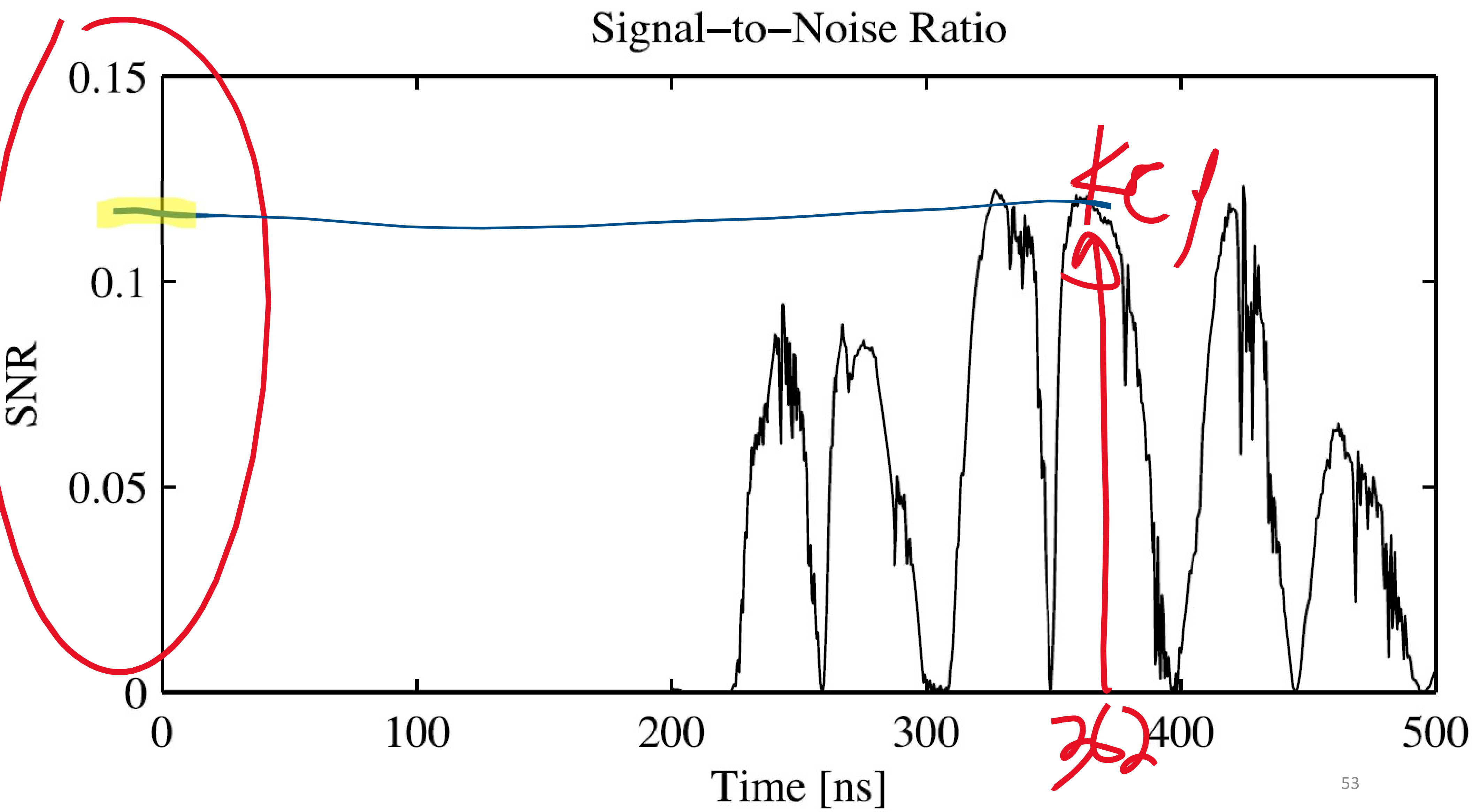
Figure 4.8. The signal levels, the standard deviation of the noise, and the SNR when attacking a uniformly distributed 1-bit data value on our microcontroller.



Mean Traces for LSB=0 and LSB=1

2 curves


Standard Deviation of Noise
330 ns
362 ns
Voltage [mV]
10
8
6
4
2
0
0
100
200
300
400
500
Time [ns]


Signal–to–Noise Ratio
0.15
0.1
0.05
0
SNR
0
100
200
300
400
500
Time [ns]
KCl
362

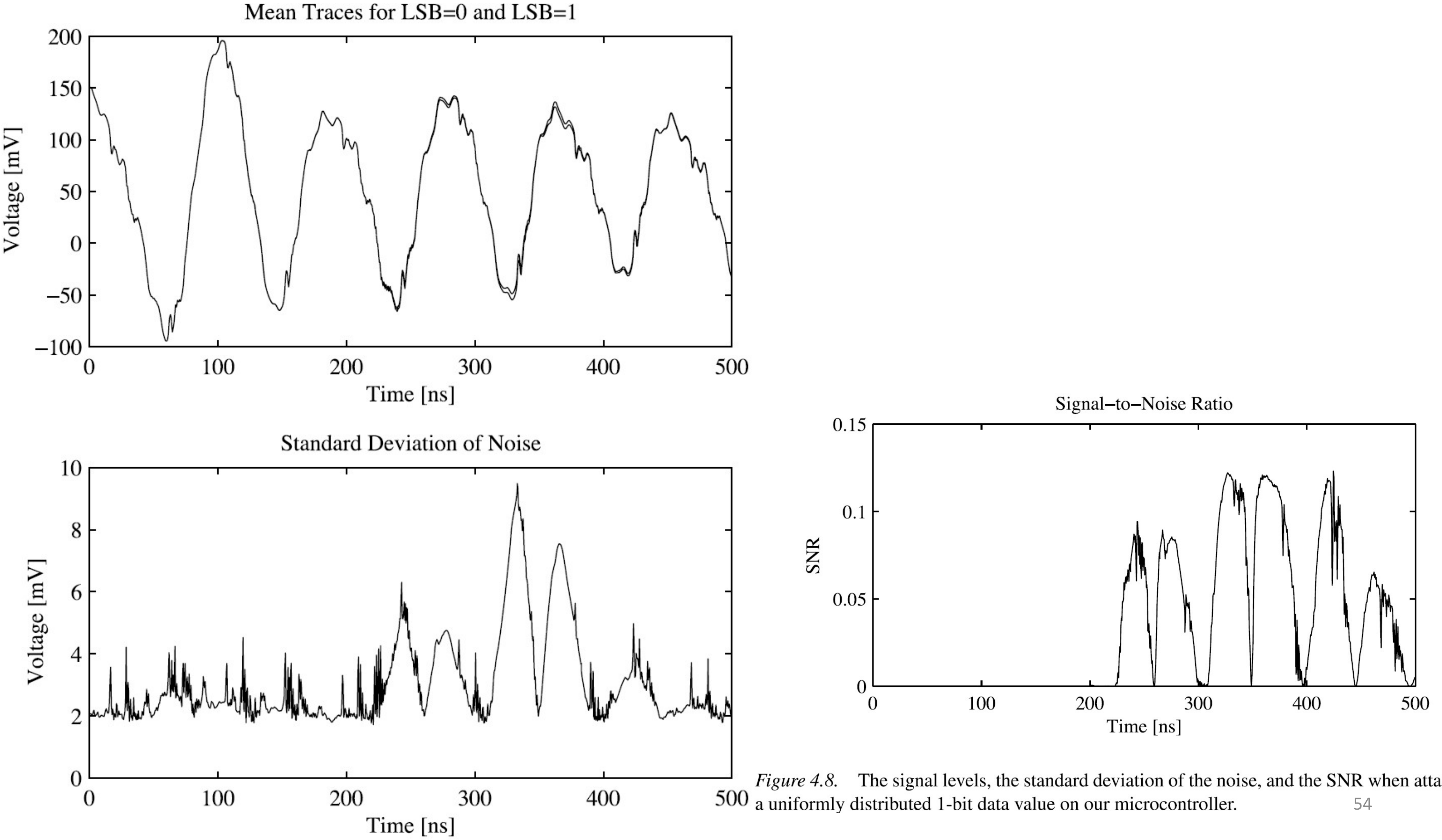


Figure 4.8. The signal levels, the standard deviation of the noise, and the SNR when attacking a uniformly distributed 1-bit data value on our microcontroller.

Comments

Comp. 4.8 (1-bit)

vs. 4.7 (8-bits)

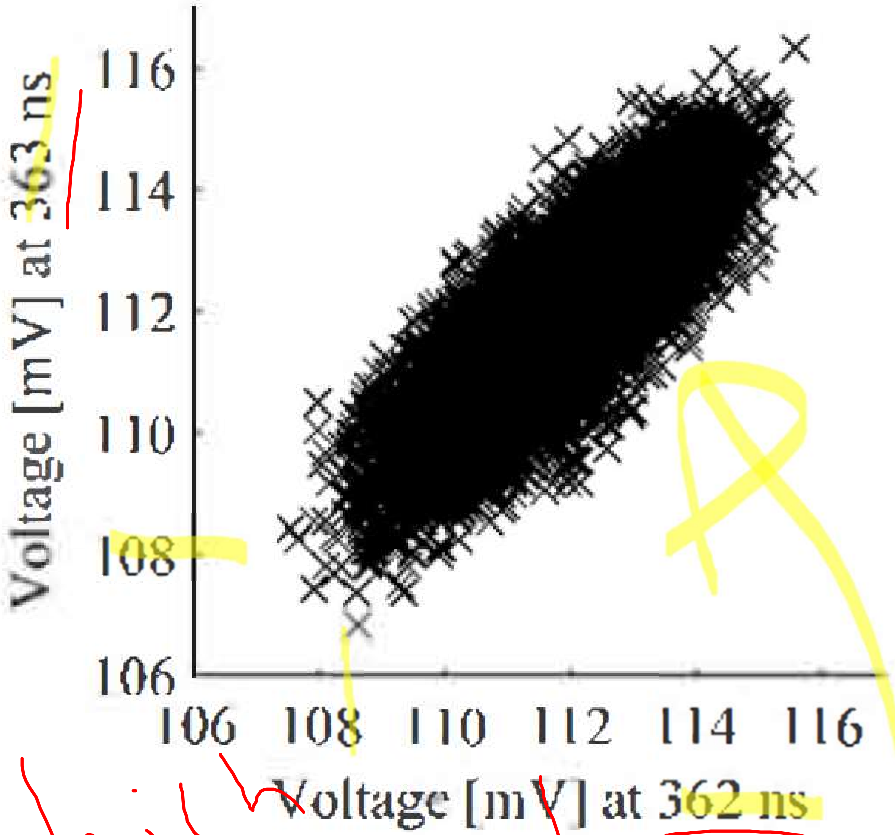
4.7 higher SNR

but 4.8 is sufficient

for a company
if 8-bit atteck
costs more
whill is prefer
8-bit (Fig 4.7) or
1-bit (Fig 4.8)

Correlation and Covariance

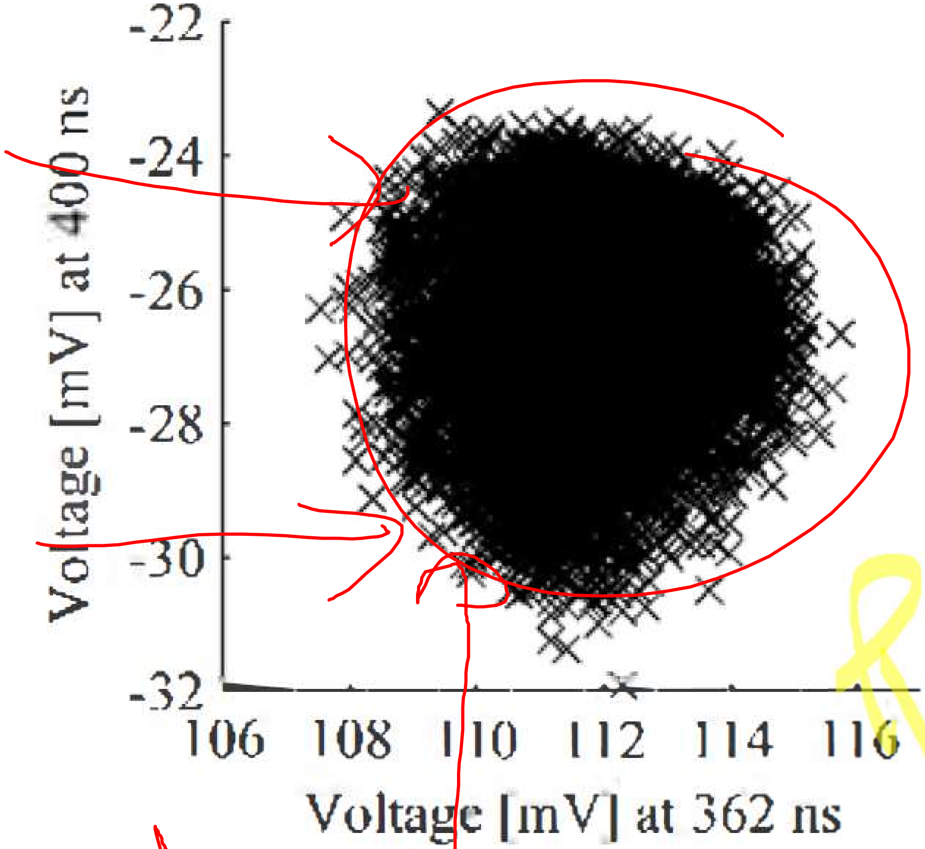
- Two points are correlated if they vary together in a related way
- Statistical measure: covariance
- *Cov(X,Y) = E[(X-E(X))*(Y-E(Y))] = E(XY) – E(X)E(Y)*
- Theoretical and empirical formulas:
- $\rho(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)*Var(Y)}}$
- $r = \frac{\sum_{i=1}^{n}(x_i - \overline{x_i})*(y_i - \overline{y_i})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x_i})^2 * \sum_{i=1}^{n}(y_i - \overline{y_i})^2}}$
- As defined, the correlation coefficient ρ varies between -1 and 1, i.e., $-1 \leq \rho \leq 1$ and also thus $-1 \leq r \leq 1$



high correlation

Figure 4.9. Scatter Plot: The power consumption at 362 ns is correlated to the power consumption at 363 ns.

***r* = 0.82**



low/no corr.

Figure 4.10. Scatter Plot: The power consumption at 362 ns is largely uncorrelated to the power consumption at 400 ns.

***r* = 0.12**